(12) UK Patent Application (19) GB (11) 2 350 034 (13) A

(43) Date of A Publication 15.11.2000

(21) Application No 9917028.4

(22) Date of Filing 20.07.1999

(30) Priority Data
(31) 9910961 (32) 11.05.1999 (33) GB

(71) Applicant(s)
3Com Corporation
(Incorporated in USA - Delaware)
5400 Bayfront Plaza, M/S 1308, Santa Clara, California 95052, United States of America

(72) Inventor(s)
Simon Peter Valentine
Lee Anthony Walker
Russell K Bulmer
Paul Robert Jones

(74) Agent and/or Address for Service
Brookes & Martin
High Holborn House, 52-54 High Holborn, LONDON, WC1V 6SE, United Kingdom

(51) INT CL[7]
H04L 12/24

(52) UK CL (Edition R)
H4P PEUX
H4K KFM

(56) Documents Cited
EP 0715435 A2 EP 0695100 A2 EP 0621706 A2
US 5737319 A US 5276789 A US 4545011 A

(58) Field of Search
UK CL (Edition R) H4K KFM KF42 , H4L LDTX , H4P PEUX PPG
INT CL[7] H04L 12/24 12/26 , H04Q 3/00
Online: WPI, EPODOC, JAPIO

(54) Abstract Title
Updating network databases

(57) A network database is initially built by automatically interrogating the network for information and by permitting an operator to manually enter information on non-managed devices which cannot be obtained through the interrogation process. A network map is visually displayed on the basis of this information. At a later time, the network is again interrogated and the new information is used to update the database. During the updating process, information originally present is not discarded if it cannot be shown to be incorrect or not present, e.g. manually entered information is preserved. The database may contain discovery, topology or sizing data and may include data relating to multinets, subnets, nodes or ports. The network may be any type of network, for example a local or wide area network (LAN or WAN) and may have cable or wireless links.

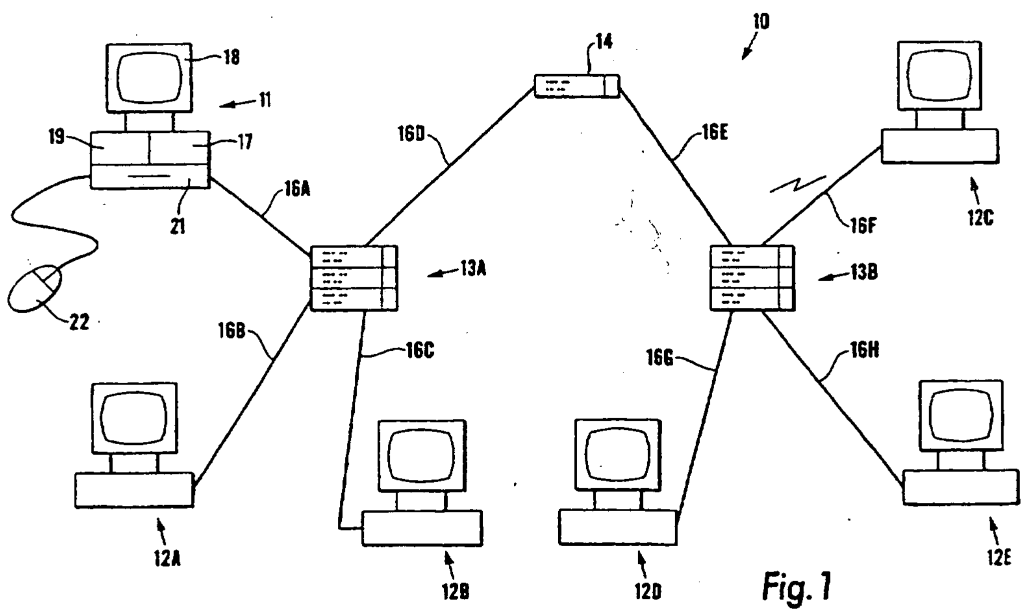
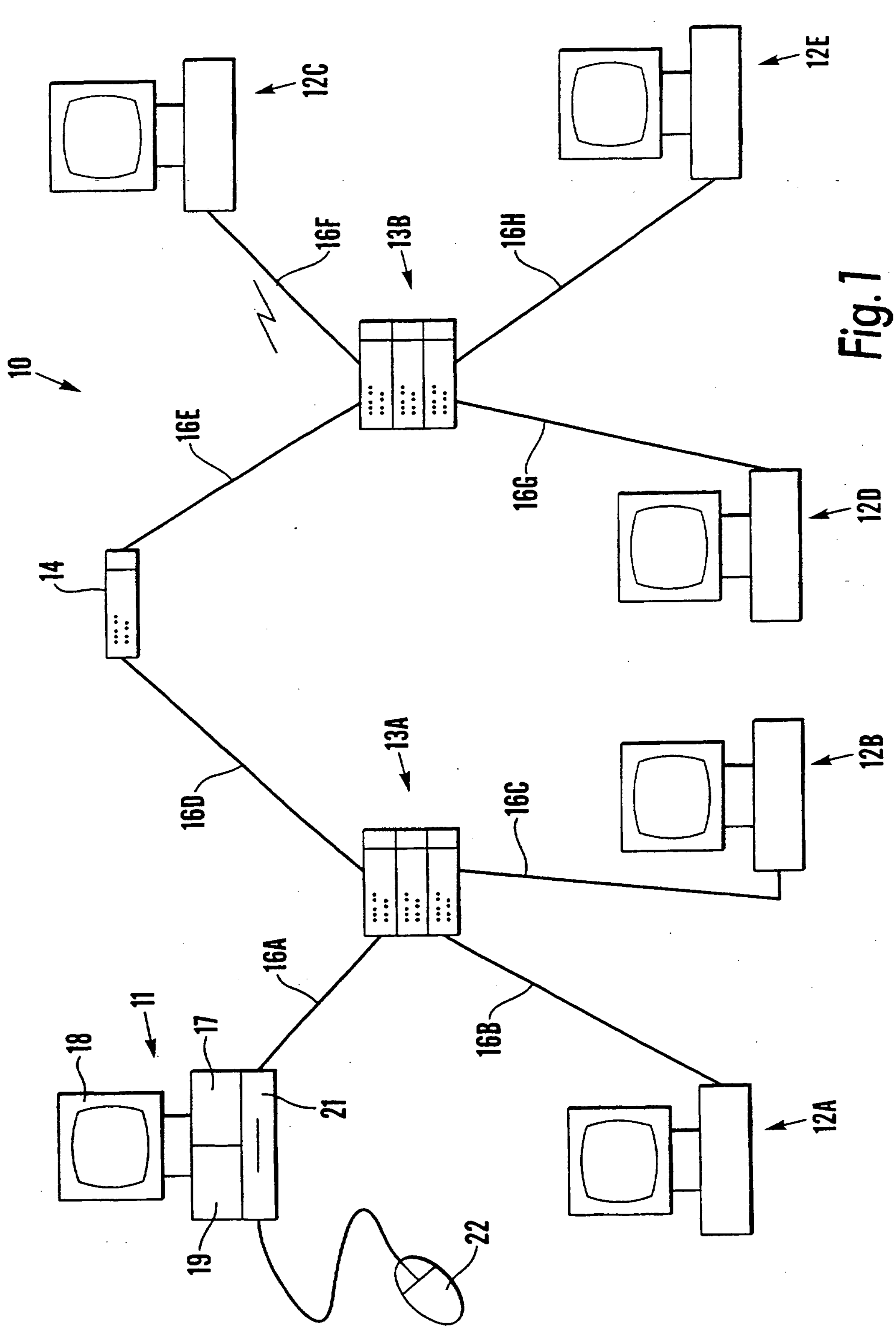


Fig. 1

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 350 034 A



10
11
12A
12B
12C
12D
12E
13A
13B
14
16A
16B
16C
16D
16E
16F
16G
16H
17
18
19
21
22

Fig. 1

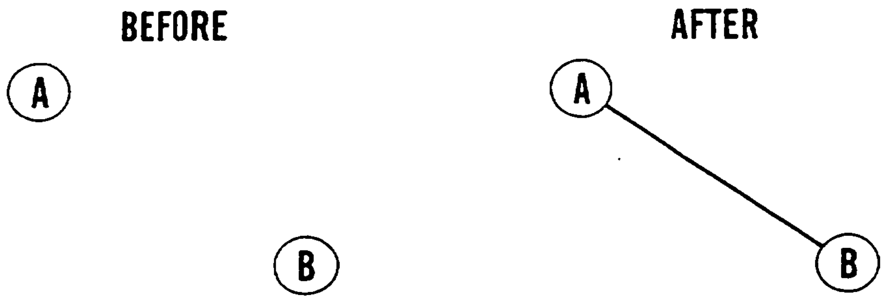


Fig.2

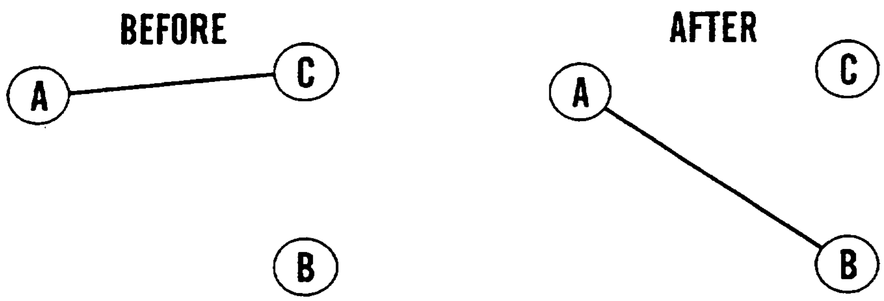


Fig.3

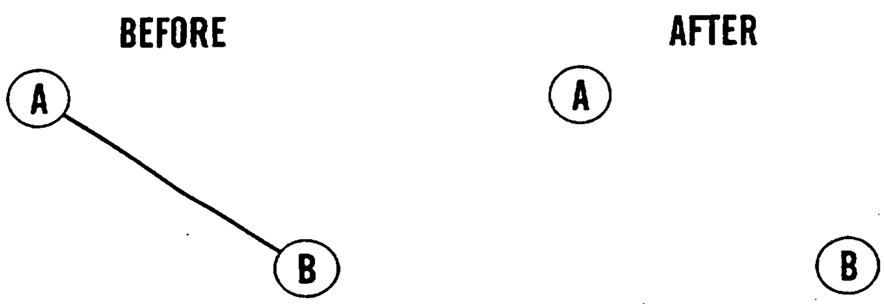


Fig.4

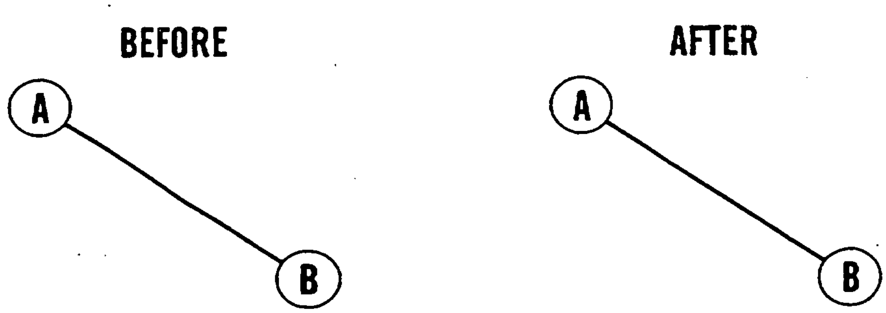


Fig.5

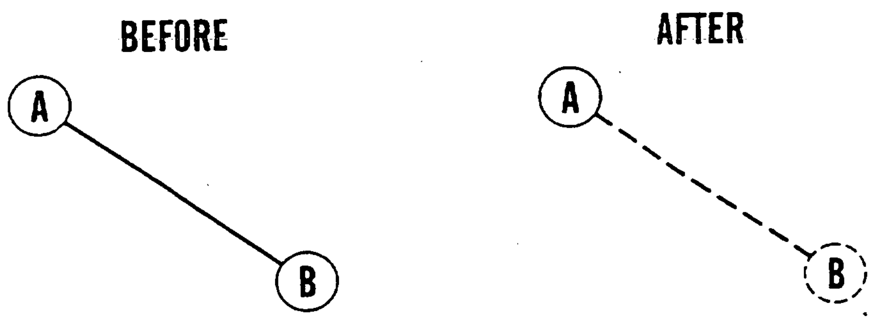


Fig.6

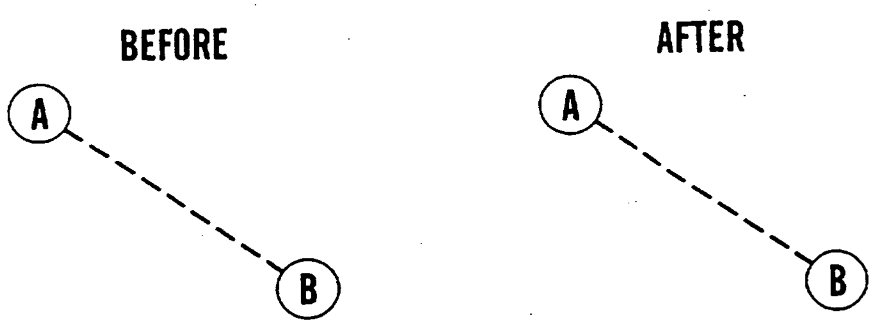


Fig.7

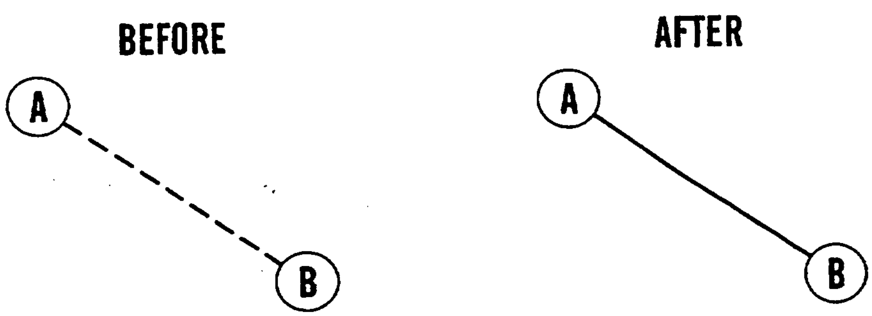


Fig.8

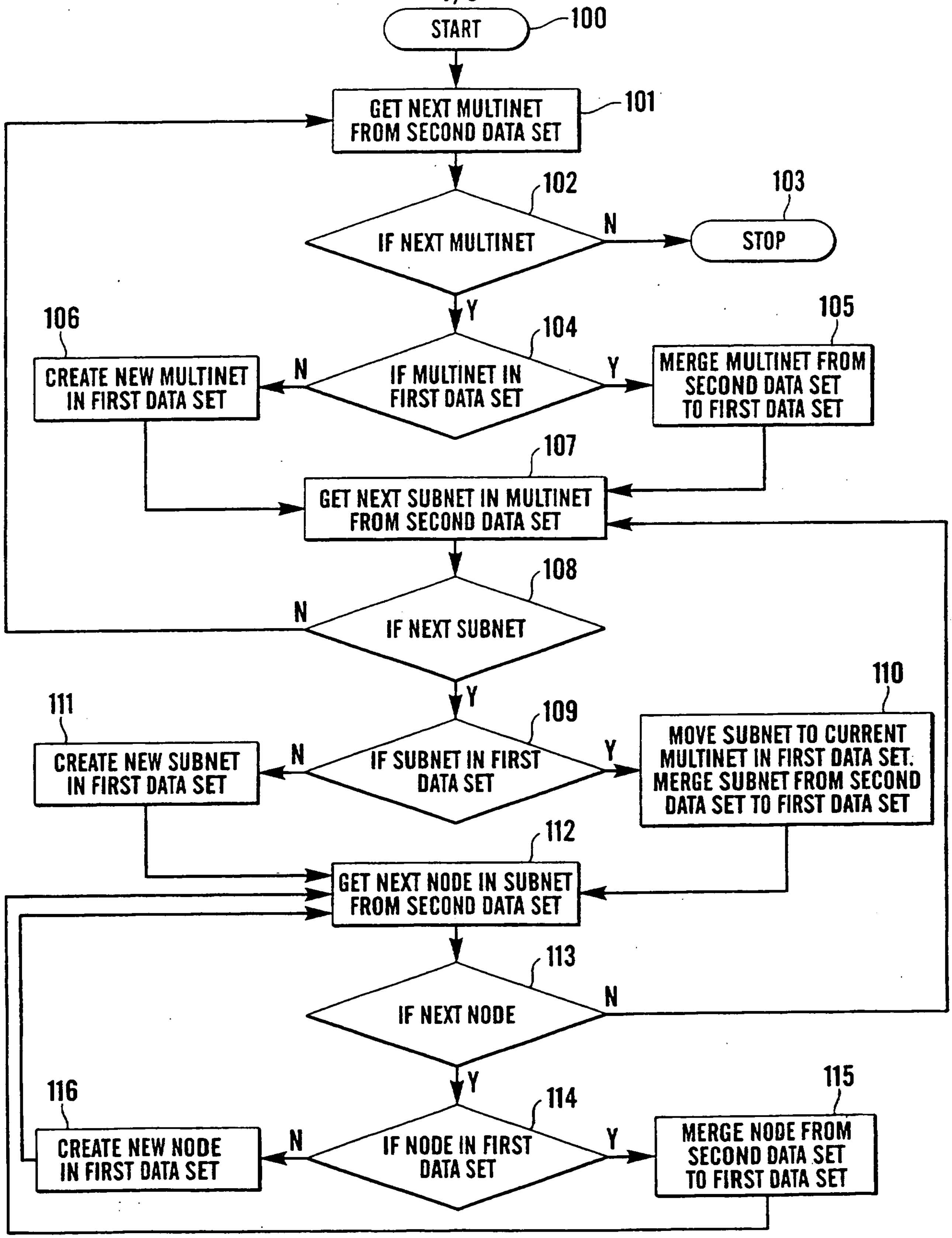

START
100
GET NEXT MULTINET FROM SECOND DATA SET
101
IF NEXT MULTINET
102
N
STOP
103
Y
IF MULTINET IN FIRST DATA SET
104
N
CREATE NEW MULTINET IN FIRST DATA SET
106
Y
MERGE MULTINET FROM SECOND DATA SET TO FIRST DATA SET
105
GET NEXT SUBNET IN MULTINET FROM SECOND DATA SET
107
IF NEXT SUBNET
108
N
Y
IF SUBNET IN FIRST DATA SET
109
N
CREATE NEW SUBNET IN FIRST DATA SET
111
Y
MOVE SUBNET TO CURRENT MULTINET IN FIRST DATA SET. MERGE SUBNET FROM SECOND DATA SET TO FIRST DATA SET
110
GET NEXT NODE IN SUBNET FROM SECOND DATA SET
112
IF NEXT NODE
113
N
Y
IF NODE IN FIRST DATA SET
114
N
CREATE NEW NODE IN FIRST DATA SET
116
Y
MERGE NODE FROM SECOND DATA SET TO FIRST DATA SET
115

Fig.9

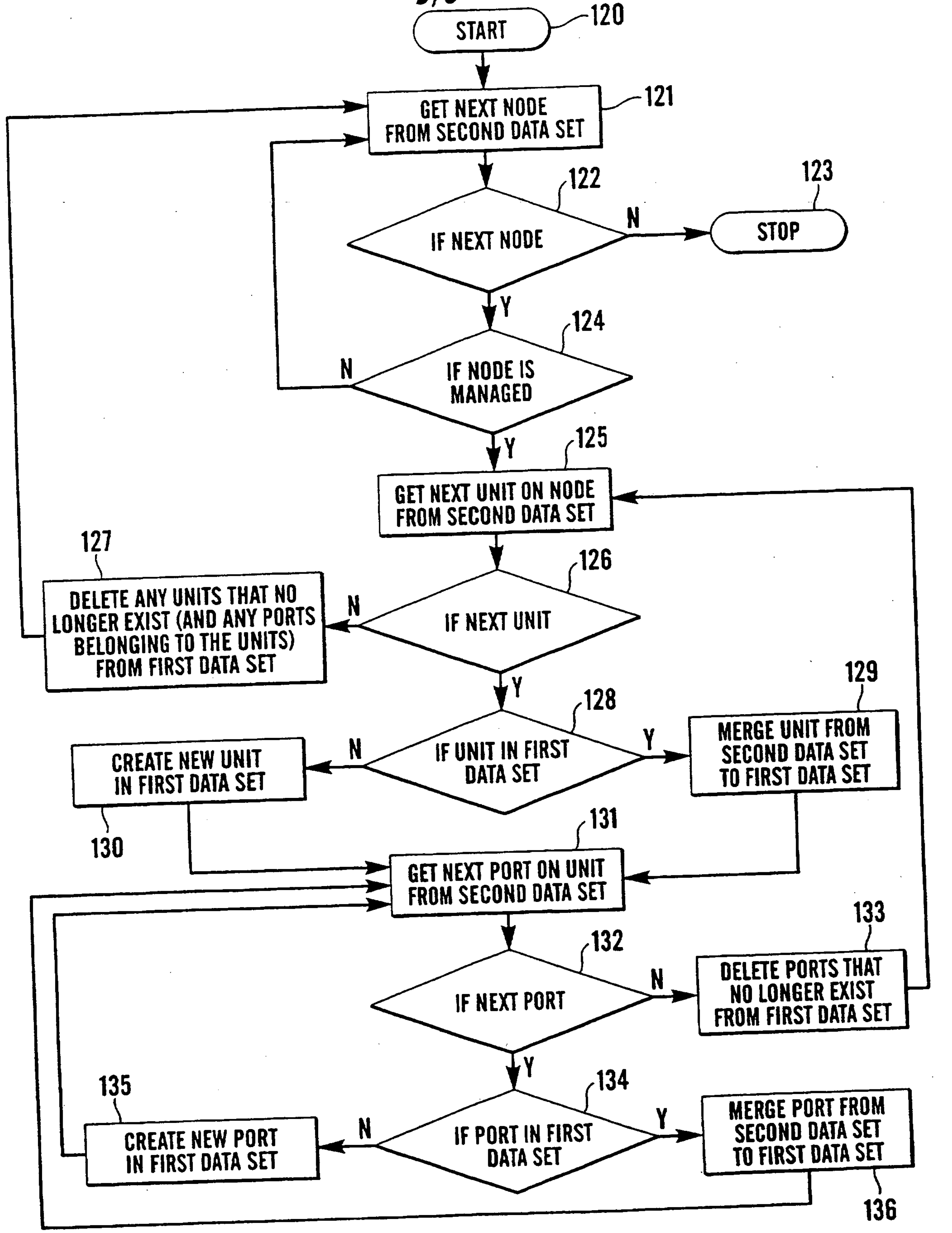

START
120
GET NEXT NODE FROM SECOND DATA SET
121
IF NEXT NODE
122
N
STOP
123
Y
IF NODE IS MANAGED
124
N
Y
GET NEXT UNIT ON NODE FROM SECOND DATA SET
125
IF NEXT UNIT
126
N
DELETE ANY UNITS THAT NO LONGER EXIST (AND ANY PORTS BELONGING TO THE UNITS) FROM FIRST DATA SET
127
Y
IF UNIT IN FIRST DATA SET
128
N
CREATE NEW UNIT IN FIRST DATA SET
130
Y
MERGE UNIT FROM SECOND DATA SET TO FIRST DATA SET
129
GET NEXT PORT ON UNIT FROM SECOND DATA SET
131
IF NEXT PORT
132
N
DELETE PORTS THAT NO LONGER EXIST FROM FIRST DATA SET
133
Y
IF PORT IN FIRST DATA SET
134
N
CREATE NEW PORT IN FIRST DATA SET
135
Y
MERGE PORT FROM SECOND DATA SET TO FIRST DATA SET
136

Fig. 10

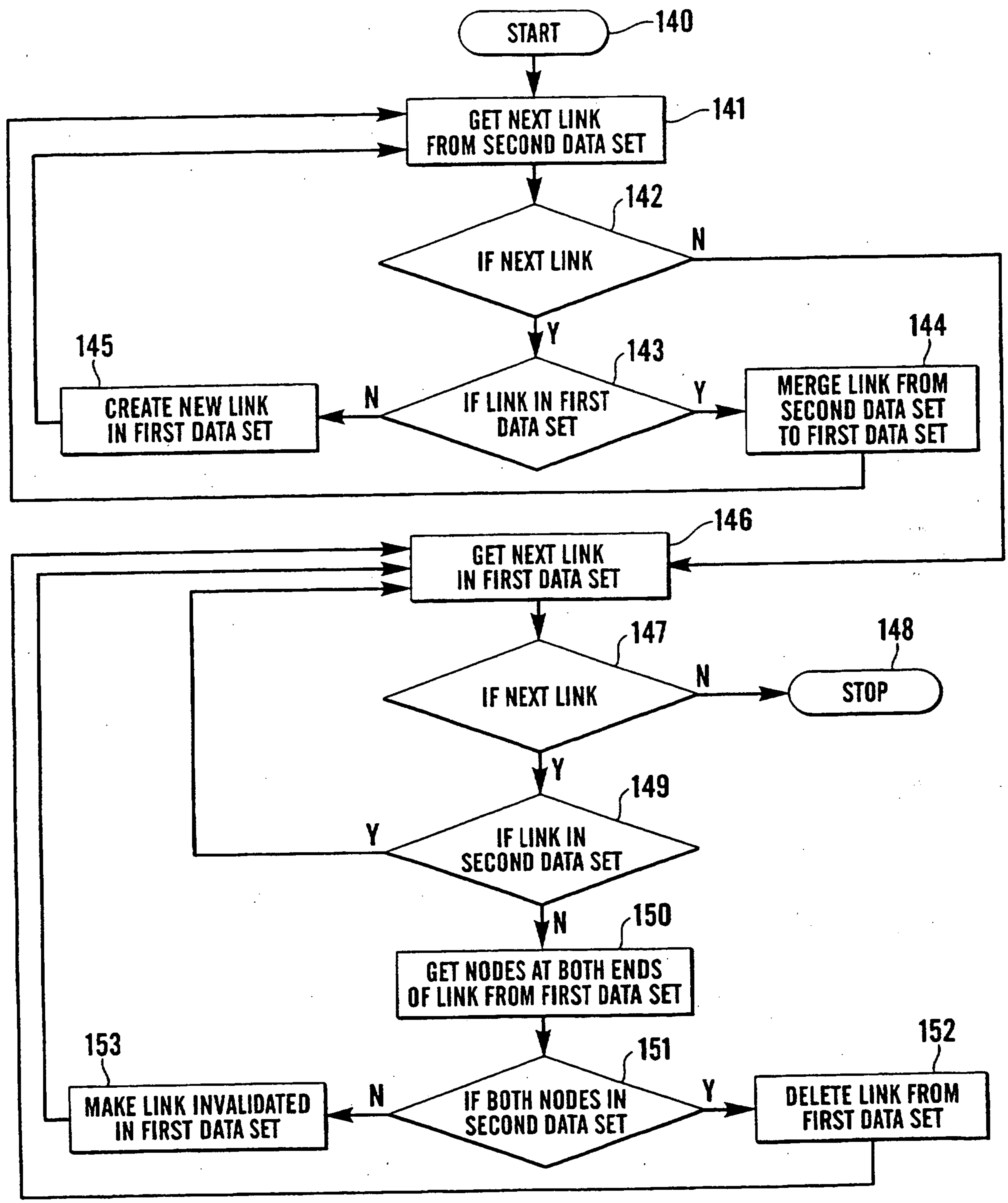

START
140
GET NEXT LINK FROM SECOND DATA SET
141
IF NEXT LINK
142
N
Y
IF LINK IN FIRST DATA SET
143
N
Y
CREATE NEW LINK IN FIRST DATA SET
145
MERGE LINK FROM SECOND DATA SET TO FIRST DATA SET
144
GET NEXT LINK IN FIRST DATA SET
146
IF NEXT LINK
147
N
STOP
148
Y
IF LINK IN SECOND DATA SET
149
Y
N
GET NODES AT BOTH ENDS OF LINK FROM FIRST DATA SET
150
IF BOTH NODES IN SECOND DATA SET
151
N
Y
MAKE LINK INVALIDATED IN FIRST DATA SET
153
DELETE LINK FROM FIRST DATA SET
152

Fig. 11

2350034

MERGING NETWORK DATABASES

BACKGROUND OF THE INVENTION

The present invention relates to merging network databases. Network databases are used in supervising a network, that is a network of electronic devices comprising, for example, workstations, personal computers, servers, hubs, routers, bridges, switches, (hereinafter referred to as devices of the network), and links between these devices which may be in the form of physical cable or wireless links. The network may be a local area network (LAN), wide area network (WAN) or other types and may operate in accordance with any desired protocol.

After such a network has been installed, it is desirable for the person appointed network manager to be able to understand the technical operation of the network. In known network management systems, the manner in which the relevant data is retrieved from the managed devices, compiled and displayed has been problematic in several respects. Firstly, the data received from each of the managed devices is simply compiled and displayed as a list of data for the user to interpret. Secondly, the data does not provide information about unmanaged devices. Thirdly, information about a given network device, such as the type of device, location of the device on the network and operating speed of the device, may be contained in different sections of the compiled data. Consequently, conventional systems are cumbersome and difficult to use.

In our co-pending UK patent applications numbers 9910843.3, 9910844.1, 9910845.8, 9910838.3, 9910837.5, 9910839.1, 9910840.9, 9910962.1 which are incorporated herein, we describe various arrangements for providing automatic interrogation of the network to thereby produce a network map which may be displayed on a visual display

unit showing the devices and links between the devices a data set with details of the network and devices. At its simplest, and where the device is a "managed" device, this interrogation uses a known protocol, such as the SNMP protocol, of the so-called 'agent' of each device which stores the devices unique MAC address, data relating to the type of device and the MAC addresses of the devices which are connected to the ports directly or indirectly.

Once this information is ascertained, it may be stored, for example, on the network manager's work station and used for various purposes. Interrogation of all the devices in the network will usually be carried out relatively infrequently.

Between these interrogations, there may be changes to the physical network itself or to the data set. For example, the network manager using his own knowledge may have manually added to the network data set non-managed devices which cannot be interrogated in the same way as managed devices, and may also have manually inserted known links between devices. Furthermore, devices and links may have been altered (ie added, removed, moved, increased or decreased in capacity). Over the course of time, therefore, the network data set may change substantially from its generation by the previous interrogation of the network.

At a later date, it may desired to interrogate the network again to verify or extend the network data set, and unless all of the earlier information is to be discarded, it is desirable to provide a system to deal with the differences between the newly generated (second) network data set, and the (first) network data set which the network manager already has. In particular, it is desirable not to discard information in the first data set which cannot be shown to be incorrect or not present (eg the information manually added by the network manager). Whilst the network manager can consider each device and link individually and compare them from the two network data sets, and make a decision, it is clearly desirable to be able to produce some kind of system or

algorithm to enable the comparison to be carried out by means, for example, of a program running an algorithm on the network manager's computer.

Thus in general terms, the present invention relates to merging one set of data representing a network, into another set of data representing the same network, but that was constructed at a different time to the first data set. The problem to be resolved is how to deal with the similarities and differences between the two sets of data.

SUMMARY OF THE INVENTION

The present invention provides a method for providing a data set relating to a network by merging a first data set relating to the network at a first time and a second data set relating to the network at a second, later, time, said first data set comprising data acquired by interrogation of the network and other data, and said second data set comprising data acquired by interrogation of the network, said method comprising:

comparing the data in the first data set with the second data set, and

updating the first data set to include additional data present in the second data set which is not present in the first data set.

In this way the information, usually the information which has been added by the network manager subsequent to the initial interrogation of the network, is preserved in creating the new data set.

Furthermore, said updating step may include the step of deleting from the first data set that data where the second data set includes information that the aspect that said data relates to is no longer present in the network.

By this method step, information which has changed and is incorrect is deleted from

the first data set.

The present invention also provides a computer program on a computer readable medium for use in providing a data set relating to a network by merging a first data set relating to the network at a first time and a second data set relating to the network at a second, later, time, said first data set comprising data acquired by interrogation of the network and other data, and said second data set comprising data acquired by interrogation of the network, said program comprising:

program means for comparing the data in the first data set with the second data set, and

program means for updating the first data set to include additional data present in the second data set which is not present in the first data set.

A preferred embodiment of the invention will now be described by way of example and with reference to the accompanying drawings in which:-

Figure 1 is a diagrammatic view of a network incorporating a preferred embodiment of the invention,

Figures 2 to 8 show a series of possible links between ports of two network devices, the left part of each drawing showing the link established in a first set of data representing a network (ie the first network map) and the right part of each drawing showing the link detected during later interrogation to produce a second, later, network map,

Figure 9 is a flow chart for merging discovery data,

Figure 10 is a flow chart for merging sizing data, and

Figure 11 is a flow chart for merging topology data.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Figure 1 there is shown a network 10 comprising a plurality of devices in the form of a network supervisor's workstation or computer 11, other workstations 12B - E, hubs 13A, 13B, and switch 14. The network is a simple network and is set out for purposes of illustration only. Other configurations and arrangements, may be used.

The devices are connected together by means of links 16A - H which may be hard wired and utilise any desired protocol, and link 16F which is a wireless link.

The network supervisor's workstation includes, in addition to a visual display unit 18, a central processing unit or signal processor 19, a selector which may be in the form of a mouse 22, a program store 21 which may comprise, for example, a CD drive, a floppy disk drive or a zip drive, and a memory 17 for storing a program which may have been loaded from the program store 21 or downloaded for example via Internet from a website.

By means which is disclosed in the co-pending patent applications referred to above, the network supervisor's computer 11 may interrogate and analyse the network, and store in the memory 17 the information relating to the devices within the network and the links between the devices. In essence, most quality devices include a so-called agent which stores information about the device such as its unique MAC number, its ID which identifies what the device is and what model type it is, how many ports it has and how they are connected, and the address to which at least some of the ports are connected. The computer 11 interrogates the agents of each device.

The information obtained by the interrogation at a first time is stored on the network supervisor's computer in the form of a first data set. This first data set will initially only include the information obtained by the interrogation but over the subsequent period of time may include further information which the network manager adds to the

data set. For example, the network manager may add information relating to non-managed devices including their links to the managed devices.

This information will comprise a first data set which includes information relating to not only the devices and links between them, but also parts of the devices, for example, units or ports (defined hereafter) and combinations of devices defined as subnets, and multinets, (which are defined hereafter).

The computer 11 may, on command from the selector 22, process signals from the memory 17 by the signal processor 19 and provide on the visual display unit 18 a network map showing each of the devices and the links therebetween.

Definitions:

Object	=	Multinet, Subnet, Node, Unit, Port, Link
Multinet	=	A group of subnets which may be linked to the remainder of the network or other multinets by means of a single switch or router
Subnet	=	A group of nodes and links which may be, for example, linked to the remainder of the network through a single switch or router
Node	=	A device connected in the network
Unit	=	A module or blade in a stack or chassis of a device
Port	=	A physical connector on a networked device to which a connection can be made
Link	=	A connection between two nodes
Discovery	=	The method used to determine details of multinet, subnet and which nodes are in each subnet
Sizing	=	The method used to determine the unit and port details on a device
Topology	=	The method used to determine details of the links between

nodes

Particular problems which need to be solved in merging two data sets on networks include:

i) Identifying objects in both sets of data, that represent the same 'real' object in the network, given that the 'real' object may change over time. Such objects include network devices, links between network devices, subnets, etc.

ii) Ensuring that objects manually added to the first data set by the user, do not get removed when merging or creating in the new set of data.

iii) Handling objects that move from one part of the network to another over time (eg between subsets)

iv) Handling changes in a network device's configuration, eg changing IP addresses, objects added/removed, etc

As examples of how the problems may be solved, we refer to Figure 2 in which the left drawing shows two devices A, B. As with all of Figures 2 to 8, the left drawing shows the network plan deduced from the original first network data set and the right drawing shows the network plan deduced by the new interrogation of the objects to form the second network data set. Thus in the original network plan, devices A and B are not connected, and after interrogation, in the new plan, devices A and B appear to be connected by a link and so in the final produced network plan, that link will be added between devices A and B. A new ID is provided.

Referring to Figure 3, there are shown three devices A, B and C, and in the original network plan devices A and C are interconnected, and when the system is interrogated to produce the new network plan, it is deduced that devices A and C are interconnected. In this case, in the new plan, the new link is added from A to B, and the link from A to C is removed because there is port conflict. However it is necessary to check the addresses (nodes) provided at all of the devices A, B and C to ensure the

veracity of the arrangement shown in the right half of Figure 3. The arrangement shown in the right half of Figure 3 is then added to the network map.

In the arrangement of Figure 4, in the original network map there is a link between devices A and B. When interrogated again, there does not appear to be a link and so this is removed.

In Figure 5, there is a link in the original network map between devices A and B, and that link is confirmed in the interrogation for the new network map, and so the link remains and the existing identity (ID) is maintained.

In Figure 6, the original network map has a link between devices A and B but on interrogation device B cannot be found (it may be present but not switched on, for example). It is thus retained as a non-validated link and device.

In Figure 7, the original network map has a user added link between devices A and B, and this is retained in the new network map. There is no way of determining whether that link is or is not present by the interrogation method employed and so one retains the relevant unverified link.

In Figure 8, once again the original map has a user added link between devices A and B, but in the new interrogation process, a link is established between A and B and so that link is turned into a validated link, that is, a non-user addition.

Figures 2 to 8 relate to the problems to be overcome in relation to different link arrangements between the first data set and the second data set. There will be similar problems to be overcome in relation to multinets, subnets, nodes, units and ports which will need to be addressed in the same or similar manner.

The flow diagrams in Figures 9, 10,11, outline the merge processes which provide the solutions set out in Figures 2 to 7 as will be set out hereafter. Figures 9, 10, 11 deal with respectively Discovery, Sizing and Topology information. For all three processes, specific match criteria have been defined for each of the object types involved, so that objects in both databases that represent the same 'real' object in the network can be identified. Subsequently this allows the process to determine whether:
a) a new object must be created (as it currently doesn't exist in the first data set), or
b) an existing object needs to be modified (as it does already exist in the first data set but has changed over time as found in the second data set), or
c) an object needs to be removed (as it does exist in the first data set, but conflicts with information in the second data set).

Note that node objects are not removed from the first data set, if they do not exist in the second data set, as there is no way of determining whether or not the said node no longer exists on the network, or is temporarily not responding.

The match criteria used by the Discovery Merge process (shown in Figure 9) to identify similar objects in both data sets, is as follows:

Multinet Match Criteria (used, for example, in step 104)	=	the multinets must contain one or more of the same subnets (refer to Subnet match criteria)
Subnet Match Criteria (used, for example, in step 109)	=	the subnets must have the same subnet IP address and use the same subnet mask
Node Match Criteria (used, for example, in step 114)	=	the nodes must have one or more MAC addresses that are the same.

The match criteria used by the Sizing Merge process (shown in Figure 10) to identify

similar objects in both data sets, is as follows:

Unit Match Criteria (used, for example in step 128)	=	i) the units must be on the same Node (refer to Node Match Criteria). ii) the units must have the same unit number.
Port Match Criteria (used, for example, in step 134)	=	i) the ports must be on the same Node (refer to Node Match Criteria) ii) the ports must be on the same Unit (refer to Unit Match Criteria). iii) the port must have the same port number.

The match criteria used by the Topology Merge process (shown in Figure 11) to identify similar objects in both data sets, is as follows:

Link Match Criteria (used, for example, in steps 143 and 149)	=	the link must connect the same two nodes(refer to Node Match Criteria)

Note that the above algorithm could be applied to any type of network, e.g. IP, IPX.

It may be noted that the process of deduction set out in Figure 2 is derived from, for example, step 145 of Figure 11.

The process of addition of a new link between A and B as set out in Figure 3 is also provided by step 145.

The process of deletion of the link between A and C as set out in Figure 3 is provided by the step 152.

The process as set out in Figure 4 is provided by step 152.

The process as set out in Figure 6 is provided by step 143 of Figure 11, and the process set out in Figure 8 is provided by step 143.

We have described how the network may be supervised. The method of the invention may be carried out under the control of the network supervisor's workstation or computer and in particular by means of a program controlling the processor apparatus of that computer or elsewhere in the system.

The program for controlling the operation of the invention may be provided on a computer readable medium, such as a CD, or a floppy disk, or a zip drive disk carrying the program or their equivalent, or may be provided on a computer or computer memory carrying the website of, for example, the supplier of the network products. The program may be downloaded from whichever appropriate source and used to control the processor to carry out the steps of the invention as described.

The program includes steps corresponding to all of the steps set out in Figures 9, 10 and 11, in particular steps 100 to 116 of Figure 9, 120 to 136 of Figure 10, and 140 to 152 of Figure 11.

The steps shown in Figure 9 are as follows:-

Step 100	Start
Step 101	Get next multinet from second data set
Step 102	Is that multinet the next multinet? If no, go to step 103, if yes, go to step 104.
Step 103	Stop
Step 104	Is that multinet in the first data set? If yes, go to step 105, if no, go to

	step 106.
Step 105	Update multinet from second data set to first data set and go to step 107.
Step 106	Create a new multinet in the first data set and go to step 107.
Step 107	Get the next subnet in the relevant multinet from the second data set.
Step 108	Is that subnet the next subnet? If yes, go to step 109, if no, return to step 101.
Step 109	Is the subnet in the first data set. If yes, go to step 110, if no, go to step 111.
Step 110	Move the subnet to the current multinet in the first data set. Update the subnet from second data set to first data set and go to step 112.
Step 111	Create new subnet in first data set.
Step 112	Get next node in subnet from second data set.
Step 113	Is the node the next node? If no, return to step 107, if yes, go to step 114.
Step 114	Is the node in first data set? If yes, proceed to step 115, if no, go to step 116.
Step 115	Update node from second data set to first data set and return to step 112.
Step 116	Create new node in first data set and return to step 112.

The steps shown in Figure 10 are as follows:-

Step 120	Start
Step 121	Get next node from second data set
Step 122	Is the node the next node? If no, go to step 123, if yes, go to step 124.
Step 123	Stop
Step 124	Is the node managed? If no, return to step 121, if yes, go to step 125.
Step 125	Get the next unit on node from the second data set.

Step 126	Is the unit the next unit? If no, go to step 127, if yes, go to step 128.
Step 127	Delete any units that no longer exist and any ports belonging to the units from the first data set and go to step 121.
Step 128	Is the unit in the first data set? If yes, go to step 129, if no, go to step 130.
Step 129	Update unit from second data set to first data set and go to step 131.
Step 130	Create new unit in first data set and go to step 131.
Step 131	Get the next port on the unit from second data set.
Step 132	Is the port the next port? If no go to step 133, if yes, go to step 134.
Step 133	Delete ports that no longer exist from the first data set and return to step 125.
Step 134	Is the port in first data set? If no, go to step 135, if yes, go to step 136.
Step 135	Create a new port in the first data set and return to step 131.
Step 136	Merge port from second data set to first data set and return to step 131.

The steps shown in Figure 11 are as follows:-

Step 140	Start
Step 141	Get next link from second data set
Step 142	Is the link the next link? If yes, go to step 143, if no, go to step 146.
Step 143	Is the link in the first data set? If yes, go to step 144, if no, go to step 145.
Step 144	Merge link from second data set to first data set and return to step 141.
Step 145	Create new link in first data set and return to step 141.
Step 146	Get the next link in the first data set.
Step 147	Is the link the next link? If no, go to step 148, if yes, go to step 149.
Step 148	Stop.
Step 149	Is the link in the second data set? If yes, return to step 146, if no, go to step 150.

Step 150	Get nodes at both ends of link from first data set.
Step 151	Are both nodes in second data set? If yes, go to step 152, if no, go to step 153.
Step 152	Delete link from first data set and return to step 146.
Step 153	Make link invalidated in first data set and return to step 146.

The invention is not restricted to the details of the foregoing example.

CLAIMS

1. A method for providing a data set relating to a network by merging a first data set relating to the network at a first time and a second data set relating to the network at a second, later, time, said first data set comprising data acquired by interrogation of the network and other data, and said second data set comprising data acquired by interrogation of the network, said method comprising:

comparing the data in the first data set with the second data set, and

updating the first data set to include additional data present in the second data set which is not present in the first data set.

2. A method as claimed in claim 1 wherein said updating step includes the step of deleting from the first data set that data where the second data set includes information that the aspect that said data relates to is no long present in the network.

3. A method as claimed in claim 1 or 2 wherein said updating step includes the step of creating new data in the first data set where additional data is present in the second set which is not present in the first set.

4. A method as claimed in claim 1 or 2 wherein said updating step includes the step of modifying new data in the first data set where additional data is present in the second set which is not present in the first set.

5. A method as claimed in any of claims 1 to 4 including an initial step of interrogation of the network at the second time to provide said second data set.

6. A method as claimed in claim 5 in which said interrogation is carried out by interrogating the agents of managed devices in the network.

7. A method as claimed in any of claims 1 to 6 in which the data includes discovery data.

8. A method as claimed in any of claims 1 to 6 in which the data includes topology data.

9. A method as claimed in any of claims 1 to 6 in which the data includes sizing data.

10. A method as claimed in any of claims 1 to 9 wherein said data includes data relating to the multinet or multinets of the network.

11. A method as claimed in any of claims 1 to 9 wherein said data includes data relating to the subnet or subnets of the network.

12. A method as claimed in any of claims 1 to 9 wherein said data includes data relating to the node or nodes of the network.

13. A method as claimed in any of claims 1 to 9 wherein said data includes data relating to the unit or units of the network.

14. A method as claimed in any of claims 1 to 9 wherein said data includes data relating to the port or ports of the network.

15. A computer program on a computer readable medium, said computer program comprising software for performing the method of any of claims 1 to 14.

16. A computer program on a computer readable medium for use in providing a data set relating to a network by merging a first data set relating to the network at a

first time and a second data set relating to the network at a second, later, time, said first data set comprising data acquired by interrogation of the network and other data, and said second data set comprising data acquired by interrogation of the network, said program comprising:

program means for comparing the data in the first data set with the second data set, and

program means for updating the first data set to include additional data present in the second data set which is not present in the first data set.

17. A computer program as claimed in claim 16 wherein said program means for updating step includes program means for deleting from the first data set that data where the second data set includes information that the aspect that said data relates to is no long present in the network.

18. A computer program as claimed in any of claims 16 to 17 in which the data includes discovery data.

19. A computer program as claimed in any of claims 16 to 18 in which the data includes topology data.

20. A computer program as claimed in any of claims 16 to 19 in which the data includes sizing data.

21. A computer program as claimed in any of claims 16 to 20 wherein said data includes data relating to the multinet or multinets of the network.

22. A computer program as claimed in any of claims 16 to 21 wherein said data includes data relating to the subnet or subnets of the network.

23. A computer program as claimed in any of claims 16 to 22 wherein said data includes data relating to the node or nodes of the network.

24. A computer program as claimed in any of claims 16 to 23 wherein said data includes data relating to the unit or units of the network.

25. A computer program as claimed in any of claims 16 to 23 wherein said data includes data relating to the port or ports of the network.

26. A network comprising a computer including a program as claimed in any of claims 16 to 25.

27. A method as claimed in claim 1 substantially as hereinbefore described.

28. A computer program as claimed in claim 16 as hereinbefore described in steps (100) to (153).

The Patent Office

INVESTOR IN PEOPLE

Application No: GB 9917028.4
Claims searched: 1-28

Examiner: Matthew Nelson
Date of search: 24 January 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): H4K (KF42, KFM); H4L (LDTX); H4P (PEUX, PPG);

Int Cl (Ed.7): H04L 12/24, 12/26; H04Q 3/00

Other: Online: WPI, EPODOC, JAPIO

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	EP 0715435 A2	(HEWLETT-PACKARD) See e.g. col. 5, line 35 - col. 7, line 51	1-5 & 7-26 at least
X	EP 0695100 A2	(AT & T) See col. 3, line 23 - col. 4, line 18; col. 5, line 40 - col. 6, line 9 & col. 6, lines 28-56	1-5 & 7-26 at least
X	EP 0621706 A2	(IBM) See col. 3, lines 14-39	1-8 12,13,15-19, 23, 24 & 26
X	US 5737319	(CROSLIN et al) See eg the abstract	1-5, 7,8,12,13, 15-19, 23, 24 & 26
X	US 5276789	(BESAW et al) See col. 2, line 51 - col. 3, line 5 & equivalent EP 457445	1-5 & 7-26 at least
X	US 4545011	(LYON et al) See the abstract and col. 8, line 5-7	1-5, 7,8,12,13, 15-19, 23, 24 & 26

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		